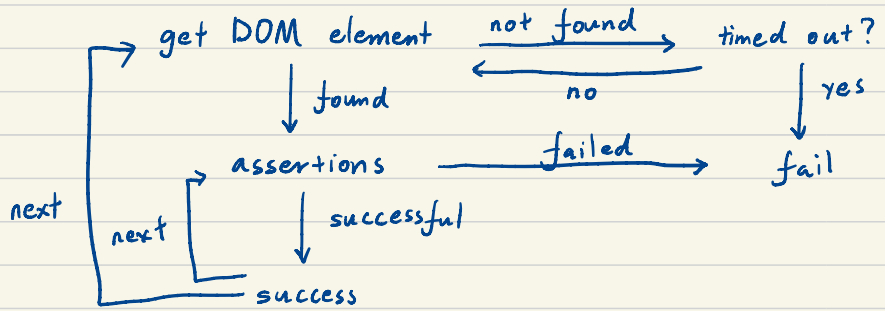
How Cypress works



- Ordering matters but the code block isn't executed synchronously
- Use .then to access the selected object yield by the cy.select or cy.contains

Alias and Not Variables

a chainer

placed on queue

```
const button = cy.get('button')
expect(button.innerText).equals('Submit')
button.click()
⋮
button.click()
```

evaluated immediately

error because button is evaluated to be a chainer

```
cy.get('button').as('button')
                .then(elem => expect(elem.innerText).equals('Submit')
cy.get('@button').click()
cy.get('@button').click()
```

Async wait won't work

- Cypress compiler chain .then to each subsequent command to implement a queue.
- Command returns a datatype called chainer that is not equivalent to promise
  e.g. no .catch method and can't use parallel Promise.all() interface

Tips

- Use data-cy attribute
- Use jQuery in browser to debug
- Use Cypress playground's selector
- Use .only to run a single test case
- add /// <reference types="Cypress" /> in documents to enable type hints

Cypress Common Methods

<button data-cy='submit'> Subit </button>

tells readers this is used for testing

- visit a page — cy.visit('url')
- select html by css selector — cy.get('[data-cy=submit]')
  - ctrl + f — cy.contains('submit', {timeout: 1000})
    - optional
- get current url — cy.url()
- interactions
  - buttons — .click() .dblclick() .rightclick()
  - checkbox — .check() .uncheck()
  - options — .select()
  - textbox — .type('text') .clear()

    key press symbols: {down arrow} {enter} {del} {option} {ctrl} {cmd}
- assertions

```
cy.get('button').should( elem => {
    expect(elem) ...
    expect ----
})
```

chai assertions

- chai assertions
  - .equals()
  - .deep.equals()
  - .match(/regex/)
  - .not

Cypress Test Suit

```
describe('Use case', () => {
    beforeEach(() => {})
    afterEach(() => {})
    it('testcase', () => {
        code
    })
})
```

Folder Structure

```
cypress
├ fixture        ← test data, retrive using cy.fixture('filename')
├ integration    ← test files
├ plugins
├ support        ← global helper functions
└ cypress.json   ← enviroment variable, retive using Cypress.env('key')
```

jQuery

- a 114 KB javascript library that standardized DOM selection across browsers
- It is now implemented natively inside browsers
- a jQuery such as $('list') is equivalent to document.querySelector('list')
  document.getElementsByClassName('list')
- jQuery also provides DOM manipulation on the selected object $('list').click().remove()

Stub the Server

- Don't overuse it
- Limited logic and lack a lot of fixtures of an actual server

```
cy.fixture('datafile').then( data =>
        cy.intercept ('GET', '/users/*', data)
)
```